

Compte rendu TP BDD-Exo

Sommaire

Introduction	1
Q1.	2
Q2.	2
Q3.	3
Q4.	4
Q5.	5
Q6.	5
Conclusion	6

Introduction

Dans ce TP, j'ai travaillé sur la création d'une base de données pour un lycée avec l'outil Looping (ou AnalyseSI). L'idée était de comprendre comment passer d'un schéma conceptuel, le MCD, qui représente les élèves et les classes, à une vraie base de données. J'ai commencé par étudier les relations et les cardinalités, puis j'ai appris à transformer ce modèle en tables SQL et à manipuler les données avec des requêtes. Ça m'a vraiment aidé à voir concrètement comment fonctionne une base de données, et à faire le lien entre la théorie et la pratique.

MCD de base



Q1.

Non. Si on regarde la cardinalité du côté de l'entité ÉLÈVE, on voit 1,1 . Cela signifie qu'un élève appartient au minimum à une classe et au maximum à une seule classe.

Q2.

```
# Modèle créé le : Tue Jan 06 13:29:38 CET 2026
ELEVE (num_ELEVE, nom_ELEVE, prenom_ELEVE, #id_classe)
classe (id_classe, libelle_classe)
```

Q3.

Requête SQL pour créer la table CLASSE

The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected. In the SQL editor, the following SQL code is entered:

```
1 CREATE TABLE CLASSE (
2     id_classe INT PRIMARY KEY,
3     libelle_classe VARCHAR(50)
4 );
```

Below the editor, a green message box indicates: "MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0012 seconde(s).)"

The results pane shows the executed SQL command:

```
CREATE TABLE CLASSE ( id_classe INT PRIMARY KEY, libelle_classe VARCHAR(50) );
```

At the bottom, there are links to edit online, edit, or create PHP code. A table below lists the columns of the CLASSE table:

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id_classe	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	2 libelle_classe	varchar(50)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus

The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected. In the SQL editor, the following SQL code is entered:

```
1 CREATE TABLE eleve (
2     num_eleve INT PRIMARY KEY,
3     nom_eleve VARCHAR(25),
4     prenom_eleve VARCHAR(25)
5 )
```

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0008 seconde(s).)

```
CREATE TABLE eleve ( num_eleve INT PRIMARY KEY, nom_eleve VARCHAR(25), prenom_eleve VARCHAR(25) );
```

[Éditer en ligne] [Éditer] [Crée le code source PHP]

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	num_eleve	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
2	nom_eleve	varchar(25)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
3	prenom_eleve	varchar(25)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus

Q4.

Requête SQL permettant d'ajouter le champ date de naissance

Serveur : 127.0.0.1 » Base de données : bdd_lycée

Structure SQL Rechercher Requête Exporter

Exécuter une ou des requêtes SQL sur la base de données « bdd_lycée » :

```
1 ALTER TABLE ELEVE
2 ADD date_naissance DATE;
```

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0008 seconde(s).)

```
ALTER TABLE ELEVE ADD date_naissance DATE;
```

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	num_eleve	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
2	nom_eleve	varchar(25)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
3	prenom_eleve	varchar(25)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
4	date_naissance	date			Oui	NULL			Modifier Supprimer Plus

Q5.

Requête SQL permettant d'ajouter les 2 élèves dans la base de données

```
1 INSERT INTO ELEVE (num_ELEVE, nom_ELEVE, prenom_ELEVE)
2 VALUES (1, 'Lapie', 'Paul');
3
4 INSERT INTO ELEVE (num_ELEVE, nom_ELEVE, prenom_ELEVE, date_naissance)
5 VALUES (2, 'Zuckerberg', 'Marc', '1984-05-14');
```

✓ 1 ligne insérée. (traitement en 0,0011 seconde(s).)

```
INSERT INTO ELEVE (num_ELEVE, nom_ELEVE, prenom_ELEVE) VALUES (1, 'Lapie', 'Paul');
```

✓ 1 ligne insérée. (traitement en 0,0004 seconde(s).)

```
INSERT INTO ELEVE (num_ELEVE, nom_ELEVE, prenom_ELEVE, date_naissance) VALUES (2, 'Zuckerberg', 'Marc', '1984-05-14');
```

	num_eleve	nom_eleve	prenom_eleve	date_naissance
<input type="checkbox"/> Éditer Copier Supprimer	1	Lapie	Paul	NULL
<input type="checkbox"/> Éditer Copier Supprimer	2	Zuckerberg	Marc	1984-05-14
<input type="checkbox"/> Éditer Copier Supprimer	1	Lapie	Paul	NULL
<input type="checkbox"/> Éditer Copier Supprimer	2	Zuckerberg	Marc	1984-05-14

Q6.

Requête SQL permettant de corriger l'erreur dans la base de données

```
1 UPDATE CLASSE
2 SET libelle_classe = 'STS1SIO'
3 WHERE id_classe = 10;
```

✓ 0 ligne affectée. (traitement en 0,0009 seconde(s).)

```
UPDATE CLASSE SET libelle_classe = 'STS1SIO' WHERE id_classe = 10;
```

Conclusion

Ce TP m'a vraiment aidé à comprendre comment fonctionne la création d'une base de données, du côté théorique jusqu'à la pratique. J'ai appris à créer des tables, à les modifier pour ajouter des informations comme la date de naissance, et à remplir les tables avec des données précises sur les élèves. C'était un exercice super utile pour me familiariser avec le SQL et mieux comprendre la gestion des données, ce qui est vraiment important pour le BTS SIO.